

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	231	((translat\$3 convert\$3 transform\$3) with (exception\$1 error\$1) with handler\$1)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/13 09:00
L2	47	1 and ((C C\$1style) near4 (exception\$1 handler\$1))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/13 09:01
L3	3	2 and (C++ near5 (application\$1 program\$1 handler\$1))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/13 09:05
L4	8	2 and ((C++ java) near5 (application\$1 program\$1 handler\$1))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/05/13 09:05

[Sign in](#)

translate "C exception" C++ handler

Results 21 - 30 of about 360 for translate "C exception" C++ handler.

▶ Web

[Images](#)[Groups](#)[News](#)[Froogle](#)[Maps](#)[more »](#)[Advanced Search](#)[Preferences](#)[.NET 247 : microsoft.public.dotnet.languages.vc Archive - October 2001](#)

AFAIK Visual C exception handling uses internally SEH. ... calling dll from managed C++ - who should allocate memory ? (3 replies, VIP) ...

www.dotnet247.com/247reference/_site/87/2001/10/2 - 27k - [Cached](#) - [Similar pages](#)

[Williams College CS334 - Programming Languages - Lecture 11](#)

begin C exception when excp_name1 => C' when excp_name2 => C" when others => C' end. When raise an exception, where do you look for handler? ...

www.cs.williams.edu/~kim/cs334/s02/Lectures/Lec11/Lec11.html - 18k -

[Cached](#) - [Similar pages](#)

[\[PDF\] Monitoring-Oriented Programming Evaluation of the Java-MOP prototype](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

Figure 5 shows the following: logic plug-ins translate formal specifications into monitors. ... reason, the violation handler will throw a specific exception, ...

www.bretagne.ens-cachan.fr/DIT/People/Claude.Jard/Stage04MIT2_Sophie.pdf -

[Similar pages](#)

[gamedev.net - Understanding C++ Exception Handling](#)

The handler for the myexceptionclass presumably does something meaningful ... I've also defined exception classes which translate directx error codes into a ...

www.dcc.unicamp.br/~ra023772/pdj/textos/exception_handling.htm - 33k - Supplemental

Result - [Cached](#) - [Similar pages](#)

[Koders - Exceptions.h](#)

Language: C++ License: GPL Copyright: (C) 1997-2000 Thingamahoochie Software ...

@brief C exception "wrapper" class for C++ try/catch ** @note : for the ...

www.koders.com/cpp/fid441BED751D92C4B92760260C391CF7A6BA5733B6.aspx - 43k -

[Cached](#) - [Similar pages](#)

[\[PDF\] Using Open Source Cores in Real Applications](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

handler routine is placed, and decides which information, about the status of the core must be ... translate from the parsed C/C++ code to OR1K assembly) ...

www.escet.urjc.es/~jcastillo/paperdcis.pdf - [Similar pages](#)

[WinMerge: an interactive diff/merge utility // Copyright \(C\) 1997 ...](#)

... @brief C exception "wrapper" class for C++ try/catch ** @note : for the ... Set the structured exception translator during the life of the object. ...

cvs.sourceforge.net/viewcvs.py/winmerge/WinMerge/Src/Exceptions.h?rev=1.2 - 5k -

[Cached](#) - [Similar pages](#)

[Slashdot | Cocoa Programming for Mac OS X, 2nd Edition](#)

If you are experienced in C++ or Java programming, Cocoa development ... I'll wait for the third edition: Protocol Handler Exploit Programming for Mac OS X. ...

books.slashdot.org/article.pl?sid=04/05/26/192224 - 123k - [Cached](#) - [Similar pages](#)

[SYSDOC NT Robert Duncan, July 1994 Porting Poplog to Windows NT ...](#)

7.4 Filename Translation ... An initial handler for the main Poplog thread is set up by renaming what has traditionally been called main in "emain.s" as ...

www.cs.bham.ac.uk/research/poplog/sysdoc/nt - 64k - [Cached](#) - [Similar pages](#)

[PDF] [ANSI/ISO C++ Professional Programmer's Handbook](#) -

File Format: PDF/Adobe Acrobat

in the catch clause of an exception **handler**); an identifier from the ... Like many UNIX compilers, it was a **translator** that first transformed **C++** code into ...

www.comms.scitech.susx.ac.uk/fft/programming/ ANSI-

ISOCPPProfessionalProgrammersHandbook.pdf - [Similar pages](#)

◀ Gooooooooooooogle ▶

Result Page: [Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [Next](#)

[Search Tips](#) | [Dissatisfied? Help us improve](#)

©2006 [Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [About Google](#)



[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: The ACM Digital Library The Guide

+C +exception +C++ +handler +catch +translate

SEARCH

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Published before April 2001

Terms used **C exception C handler catch translate**

Found 138 of 118,990

Sort results
by

▾

[Save results to a Binder](#)

Display
results

▾

[Search Tips](#)

Open results in a new window

Try an [Advanced Search](#)

Try this search in [The ACM Guide](#)

Results 1 - 20 of 138

Result page: 1 2 3 4 5 6 7 [next](#)

Relevance scale

1 Optimizing away C++ exception handling

Jonathan L. Schilling

August 1998 **ACM SIGPLAN Notices**, Volume 33 Issue 8

Publisher: ACM Press

Full text available: [pdf\(899.59 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

A high performance implementation of C++ exception handling is crucial, because exception handling overhead is distributed across all code. The commonly-used table-driven approach to implementing exception handling can be augmented by an optimization that seeks to identify functions for which (contrary to first appearance) no exception handling tables need be generated at all. This optimization produces modest but useful gains on some existing C++ code, but produces very significant size and spe ...

Keywords: C++, benchmarks, compiler, exception handling, optimization

2 Efficient Java exception handling in just-in-time compilation

SeungIl Lee, Byung-Sun Yang, Suhyun Kim, Seongbae Park, Soo-Mook Moon, Kemal Ebcioğlu, Erik Altman

June 2000 **Proceedings of the ACM 2000 conference on Java Grande**

Publisher: ACM Press

Full text available: [pdf\(641.73 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

3 A single intermediate language that supports multiple implementations of exceptions

Norman Ramsey, Simon Peyton Jones

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation PLDI '00**, Volume 35 Issue 5

Publisher: ACM Press

Full text available: [pdf\(900.75 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present mechanisms that enable our compiler-target language, C--, to express four of the best known techniques for implementing exceptions, all within a single, uniform framework. We define the mechanisms precisely, using a formal operational semantics.

We also show that exceptions need not require special treatment in the optimizer; by introducing extra dataflow edges, we make standard optimization techniques work even on programs that use exceptions. Our approach clarifies the design s ...

4 Java bytecode to native code translation: the caffeine prototype and preliminary results

Cheng-Hsueh A. Hsieh, John C. Gyllenhaal, Wen-mei W. Hwu

December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture**

Publisher: IEEE Computer Society

Full text available:  pdf(1.03 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Java bytecode language is emerging as a software distribution standard. With major vendors committed to porting the Java run-time environment to their platforms, programs in Java bytecode are expected to run without modification on multiple platforms. These first generation run-time environments rely on an interpreter to bridge the gap between the bytecode instructions and the native hardware. This interpreter approach is sufficient for specialized applications such as Internet browsers wher ...

5 Efficient and precise modeling of exceptions for the analysis of Java programs

 Jong-Deok Choi, David Grove, Michael Hind, Vivek Sarkar

September 1999 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1999 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering PASTE '99**, Volume 24 Issue 5

Publisher: ACM Press

Full text available:  pdf(1.16 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Factored Control Flow Graph, *FCFG*, is a novel representation of a program's intraprocedural control flow, which is designed to efficiently support the analysis of programs written in languages, such as Java, that have frequently occurring operations whose execution may result in exceptional control flow. The FCFG is more compact than traditional CFG representations for exceptional control flow, yet there is no loss of precision in using the FCFG. In this paper, we introduce the FCFG r ...

6 Ada, C, C++, and Java vs. the Steelman

 David A. Wheeler

July 1997 **ACM SIGAda Ada Letters**, Volume XVII Issue 4

Publisher: ACM Press

Full text available:  pdf(1.57 MB)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This paper compares four computer programming languages (Ada95, C, C++, and Java) with the requirements of "Steelman", the original 1978 requirements document for the Ada computer programming language. This paper provides a view of the capabilities of each of these languages, and should help those trying to understand their technical similarities, differences, and capabilities.

7 A comparison of Ada and C++

 L. S. Tang

December 1992 **Proceedings of the conference on TRI-Ada '92**

Publisher: ACM Press

Full text available:  pdf(1.08 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

 [Targeting GNAT to the Java virtual machine](#)

Cyrille Comar, Gary Dismukes, Franco Gasperoni

November 1997 **Proceedings of the conference on TRI-Ada '97**

Publisher: ACM Press

Full text available:  [pdf\(1.72 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

9 A study on exception detection and handling using aspect-oriented programming

 Martin Lippert, Cristina Videira Lopes

June 2000 **Proceedings of the 22nd international conference on Software engineering**

Publisher: ACM Press

Full text available:  [pdf\(128.27 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Aspect-Oriented Programming (AOP) is intended to ease situations that involve many kinds of code tangling. This paper reports on a study to investigate AOP's ability to ease tangling related to exception detection and handling. We took an existing framework written in Java™, the JWAM framework, and partially reengineered its exception detection and handling aspects using AspectJ™, an aspect-oriented programming extension to Java. We found that AspectJ supported impleme ...

Keywords: aspect-oriented programming, contracts, exceptions

10 An out-of-order execution technique for runtime binary translators

 Bich C. Le

October 1998 **ACM SIGOPS Operating Systems Review , ACM SIGPLAN Notices , Proceedings of the eighth international conference on Architectural support for programming languages and operating systems ASPLOS-VIII**, Volume 32 , 33 Issue 5 , 11

Publisher: ACM Press

Full text available:  [pdf\(1.04 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A dynamic translator emulates an instruction set architecture by translating source instructions to native code during execution. On statically-scheduled hardware, higher performance can potentially be achieved by reordering the translated instructions; however, this is a challenging transformation if the source architecture supports precise exception semantics, and the user-level program is allowed to register exception handlers. This paper presents a software technique which allows a translat ...

11 A generalization of exceptions and control in ML-like languages

 Carl A. Gunter, Didier Rémy, Jon G. Riecke

October 1995 **Proceedings of the seventh international conference on Functional programming languages and computer architecture**

Publisher: ACM Press

Full text available:  [pdf\(1.34 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

12 ASHs: application-specific handlers for high-performance messaging

Deborah A. Wallach, Dawson R. Engler, M. Frans Kaashoek

August 1997 **IEEE/ACM Transactions on Networking (TON)**, Volume 5 Issue 4

Publisher: IEEE Press

Full text available: Additional Information:

 pdf(174.62 KB)[full citation](#), [references](#), [index terms](#)

Keywords: computer networks, dynamic code generation, modular computer systems, operating systems, protocols, software protection, user-level networking

13 ASHs: Application-specific handlers for high-performance messaging 

 Deborah A. Wallach, Dawson R. Engler, M. Frans Kaashoek
August 1996 **ACM SIGCOMM Computer Communication Review , Conference proceedings on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '96**, Volume 26 Issue 4

Publisher: ACM Press

Full text available:  pdf(174.50 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Application-specific safe message handlers (ASHs) are designed to provide applications with hardware-level network performance. ASHs are user-written code fragments that safely and efficiently execute in the kernel in response to message arrival. ASHs can direct message transfers (thereby eliminating copies) and send messages (thereby reducing send-response latency). In addition, the ASH system provides support for dynamic integrated layer processing (thereby eliminating duplicate message ...)

14 Hardware and software support for efficient exception handling 

 Chandramohan A. Thekkath, Henry M. Levy
November 1994 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the sixth international conference on Architectural support for programming languages and operating systems ASPLOS-VI**, Volume 29 , 28 Issue 11 , 5

Publisher: ACM Press

Full text available:  pdf(1.44 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Program-synchronous exceptions, for example, breakpoints, watchpoints, illegal opcodes, and memory access violations, provide information about exceptional conditions, interrupting the program and vectoring to an operating system handler. Over the last decade, however, programs and run-time systems have increasingly employed these mechanisms as a performance optimization to detect normal and expected conditions. Unfortunately, current archi ...

15 Tolerating exceptions in workflows: a unified framework for data and processes 

 Alex Borgida, Takahiro Murata
March 1999 **ACM SIGSOFT Software Engineering Notes , Proceedings of the international joint conference on Work activities coordination and collaboration WACC '99**, Volume 24 Issue 2

Publisher: ACM Press

Full text available:  pdf(1.27 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Practical workflow systems need to be able to tolerate deviations from the initial process model because of un-anticipated situations. They should also be able to accommodate deviations in the format of the forms and data being manipulated. We offer a framework for treating both kinds of deviations uniformly, by applying ideas from programming languages (with workflow agents as potential on-line exception handlers) to workflows that have been reified as objects in classes with special attributes ...

Keywords: deviations, exception handling, reified process model, safety

16 A high performance Erlang system

Erik Johansson, Mikael Pettersson, Konstantinos Sagonas

September 2000 **Proceedings of the 2nd ACM SIGPLAN international conference on Principles and practice of declarative programming**

Publisher: ACM Press

Full text available: [pdf\(320.62 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**17 Type-based analysis of uncaught exceptions**

François Pessaux, Xavier Leroy

January 1999 **Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Publisher: ACM Press

Full text available: [pdf\(1.92 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**18 Practicing JUDO: Java under dynamic optimizations**

Michał Cierniak, Guei-Yuan Lueh, James M. Stichnoth

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation PLDI '00**, Volume 35

Issue 5

Publisher: ACM Press

Full text available: [pdf\(190.06 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A high-performance implementation of a Java Virtual Machine (JVM) consists of efficient implementation of Just-In-Time (JIT) compilation, exception handling, synchronization mechanism, and garbage collection (GC). These components are tightly coupled to achieve high performance. In this paper, we present some static and dynamic techniques implemented in the JIT compilation and exception handling of the Microprocessor Research Lab Virtual Machine (MRL VM), ...

19 Features of the GNU Ada runtime library

E. W. Giering, Frank Mueller, T. P. Baker

November 1994 **Proceedings of the conference on TRI-Ada '94**

Publisher: ACM Press

Full text available: [pdf\(978.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Gnu Ada Runtime Library (GNARL) is being developed to support Ada 9X tasking for the Gnu NYU Ada Translator (GNAT). Together, they form a portable, freely distributable Ada 9X translation system. GNARL and GNAT communicate through a well-defined procedural interface, facilitating their independent development. Among the design goals of this translation system are portability, interoperability with other languages (in particular C), efficiency, and user extensibility. This paper ...

20 A formal framework for the Java bytecode language and verifier

Stephen N. Freund, John C. Mitchell

October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99**, Volume 34 Issue 10

Publisher: ACM Press

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

Full text available: [!\[\]\(7e49c700e4adaed94ad5398cf2e7059e_img.jpg\) pdf\(1.93 MB\)](#)

[terms](#)

This paper presents a sound type system for a large subset of the Java bytecode language including classes, interfaces, constructors, methods, exceptions, and bytecode subroutines. This work serves as the foundation for developing a formal specification of the bytecode language and the Java Virtual Machine's bytecode verifier. We also describe a prototype implementation of a type checker for our system and discuss some of the other applications of this work. For example, we show how to exte ...

Results 1 - 20 of 138

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [!\[\]\(0ac73c45806a78de248a19d9a2dbe7a6_img.jpg\) Adobe Acrobat](#) [!\[\]\(8d9d43d9f7500adebe1b57d821f10a8c_img.jpg\) QuickTime](#) [!\[\]\(107e04cbc96859c8386dd3148020c677_img.jpg\) Windows Media Player](#) [!\[\]\(cf35150433f6591f40a9bc6e2fc46d01_img.jpg\) Real Player](#)